REMARKS

Claims 1, 3-10 are rejected under 25 U.S.C. 103(a) as being unpatentable over Jian et al (6,567,980) and Takemoto (6,335,742). The applicants respectfully traverse this rejection on grounds that the Jain et al. reference does not teach or suggest selected points for which it is cited in Office Action of September 27, 2003.

I. Claim 1:

The following reviews various points made in the Office Action of September 22, 2003 about what is shown in the Jain et al. reference and responds to each point:

1. a GUI adapted to browse pictures stored in a database (column 2 lines 10-20, column 4 lines 20-35), including: a main level display providing links to other display levels (Fig. 2)

Jain et al. does not describe at Fig. 2, a main level display providing links to other display levels. Instead, what is shown in Fig. 2 is described column 4 lines 21-40 of the Jain et al. patent. These lines state follows:

> *FIG. 2 depicts an example user interface that is representative of the type of graphical user interface (GUI) than could be built around the Video Engine shown in FIG. 9. In FIG. 2, the Video Cataloger user interface is contained in a window 170. The main controls are exposed as menus and a tool bar 182. A panel 172 displays the live video being digitized, with play, stop, etc. controls that interact remotely with the analog source via a deck controller 240 (FIG. 3). Keyframes extracted during the capture process are displayed in a panel 176, while the corresponding close-caption text and timecodes are displayed in a panel 178. A panel 184 displays the user-defined clip annotations, created by marking in- and out-points. The columns 186 and 188 display the in- and out-time codes for the marked clip, respectively, while the remaining columns 190, 192, 194 are an example of a user defined schema of labels to describe the clip. Finally, at the bottom of the window 170 is a timeline 180 that depicts the total time of the capture session, with a highlighted section corresponding to the currently selected range of keyframes.*

There is no teaching in the cited section or in that Fig. 2 comprises a main level displays that information providing links to other display levels. In fact, it

appears that main level shown in Fig. 2 is configured to provide all at once access to information regarding video clips, and selected key frames.

2. a picture content iconic region having icons representing pictures according to predefined content categories and picture metadata Figs. 14, 17 column 14 lines 27-65);

According to the Jain et al. reference, Fig. 14 is "a flowchart of the feature extraction process shown in FIG. 13" Figs. 13 and 14 are described at columns 11 lines 52-65 and and column 12 lines 1-57. These lines state as follows:

> *FIG. 13 details the metadata capture process 776 which is an important activity of the Video Engine 440 of FIG. 9. The metadata capture process 776 was first introduced in FIG. 12.*
>
> *The capture process 776 begins with the scheduling of a system timer event in step 804 set to go off [fraction (1/30)] of a second in the future. The control flow of the process 776 immediately proceeds to the Get Event step 806 where other system events (besides the timer event) may be processed. When an event occurs, control passes to the Event Dispatcher 808 which decides if the event is one of the two types of events: a normal GUI event, or the scheduled timer event.*
>
> *For a GUI event, the event is first inspected in step 812 to determine if it is an End Capture event, in which case the capture process loop terminates. If not, processing proceeds to step 816 to handle the GUI event (such as keystroke, window resized, etc.). Some GUI events may generate metadata (if the user marked a video clip), which is determined in step 818. If metadata (a video clip) was in fact generated, that metadata is committed to the Metadata Track Index Manager 530 (FIG. 9) during step 820. This also necessitates a GUI redraw, so the affected parts of the GUI are marked for Redraw in step 822.*
>
> *If the event dispatched in 808 is the timer event., this signifies that feature extraction of metadata from the video signals is to take place at a feature extraction process 810. The details of the feature extraction process 810are provided in conjunction with FIG. 14. Once feature extraction is complete, control moves to step 804 where the next timer event is scheduled.*
>
> *This flow of activity is tied to the event model of the operating system under which the software application is running. The flow that is shown is an event model that is typical of a Windows MFC-based application. Other operating system platforms, such as Unix, have event models that differ somewhat. The event model*

illustrates how the feature extraction process fits into an application event framework. Note that, in the depicted embodiment, the Get Event task 806 is a call out to Windows MFC, which processes Redraw Events by calling the Redraw method of the appropriate GUI elements directly (this process diagram does not "call" the Redraw methods directly). Note that it is acceptable if feature extraction takes more than [fraction (1/30)] second.
14. Feature Extraction--Flowchart

FIG. 14 details the feature extraction process 810, which is an important aspect of the present invention, relying on the innovative architecture of FIG. 9.

The feature extraction process 810 begins at a start step 842 and proceeds to step 844 where the current time code is obtained by module 502 of FIG. 9. This time code is used by all feature extractors to time-stamp the metadata they extract. Next, all digital media is captured in step 846 by modules 504, 506, and 508 of FIG. 9. This digital media is then passed on to the Feature Extractor Framework 510 (FIG. 9) for processing. The Feature Extractor Framework 510 spawns a process thread 850 for each feature extractor. Each feature extractor processes the digital media in step 852 in whatever way it desires, for example, extract a keyframe, classify the audio signal, etc. In certain cases, but not all, some metadata will be generated from this process. Step 854 determines if this is the case, and if so, the metadata is passed to the Metadata Track Index Manager 530(FIG. 9) during step 856. Since metadata is usually displayed in real-time in the GUI, the GUI is marked for redraw in step 858. One particular exemplary feature: extractor for video keyframes is described in the pending U.S. patent application entitled "Key Frame Selection" filed on Jun. 6, 1997.

When all feature extractor threads complete, as determined at wait (synchronization) step 862, control is returned to the capture metadata process at end step 864.

What appears to be described herein the steps of extracting metadata and features from the video content. As is stated at columns 12 lines 50-52 "since metadata is usually displayed in real-time in the GUI, the GUI is marked for redraw in step 858." Thus, at the conclusion of the processes described in Fig. 13 and 14, what is disclosed in the Jain et al. reference is amending the displayed listing of metadata. There is no discussion in these lines of providing a content iconic region having icons representing pictures according to predefined content categories and picture metadata. It is also noted that there is no discussion of moving to another display level instead of the current level of the metadata is updated.

Fig. 17 is described as "an exemplary screen display seen as an output of the HTML output filter process of FIG. 16 while using a client browser for the multimedia cataloger system shown in FIG. 1." As described at columns 12 lines 60-65 and and one columns 13 lines 1-32 Fig. 17 shows the following:

> *The Output Filter Manager 560 (FIG. 8) may utilize a HTML output filter 564 in one embodiment. Referring to FIG. 15, elements of FIGS. 1, 2 and 9 are shown together as utilized in generating HTML output. The user may invoke a GUI command such as the "Save-As" command on the "File" menu 553, which in turn provides a list of output filter choices (HTML, Real Networks SMIL, XML, custom, etc.). When the HTML filter 564 is invoked, it accesses the metadata in the Metadata Track Index Manager 530 and processes it into HTML form in a browser window 916 (FIG. 17), which also involves keyframe images in a keyframe frame 176 (FIG. 2) or 904 (FIG. 17), and the digital video 142 (FIG. 1) or as seen in a video frame 896 (FIG. 17). For instance, hyperlinks may be formed from displayed keyframes to video sequences. The digital video 142 may or may not be served by a content server 140. For instance, it could be a simple file on the file system of the client computer or, say, a networked mass storage device visible to the computer.*
>
> *Some key features of the Video Cataloger HTML output are:*
>
> *a. The HTML files used to generate the display in the browser window 916 (FIG. 17) are completely stand-alone, internally linked HTML, such that no Web server is required. Exemplary HTML files are provided in the Appendix and are described in conjunction with FIG. 17 below.*
>
> *b. It incorporates play-back of digital video 142 from a file or from a video server 140. That is, the digital video may be streamed directly to the browser, or it may simply be played from a local file on disk. The stand- alone aspect is strengthened when the digital video is a local file. This way, all of the content (HTML, keyframes, digital video) could be packaged up, compressed, and e-mailed to someone.*
>
> *c. All metadata is cross-referenced/cross-linked based on time-codes.*
>
> *d. Digital video is independent of the HTML representation—any digital video source can be linked into the playback frame.*

Thus, as is shown in these lines, the window provided in Fig. 17 is a completely stand alone HTML browser window with video clips linked to the HTML browser window such that the video clips can be accessed without need to access a web

server. There is no discussion in this section however, that Fig. 17 has a picture content iconic region having icons representing pictures according to predefined content categories and picture metadata. Instead, Fig. 17 is merely an HTML version of what is shown in Fig. 2, which as has been discussed above.

As noted in the Office Action, Fig. 17 is further described at column 14, lines 27-65, these lines state as follows:

> *17. Example HTML Output--Screen Shot*
>
> *Referring to FIGS. 16 and 17, a screen shot of the HTML output as seen at a client browser and as generated by the HTML output process 890 (FIG. 16) will be described. Element 896 corresponds to the video frame in the upper left portion of the screen display. Element 904 corresponds to the keyframe frame in the lower left portion of the screen display. Element 908 corresponds to the cc-text frame in the lower right portion of the screen display. Element 912 corresponds to the clip frame in the upper right portion of the screen display. Element 916 corresponds to the whole browser window. As with most browsers, including Microsoft Explorer and Netscape Navigator, if the displayable page is larger than the physical display, the browser will cause the page to be scrolled. Video data is retrieved by sending a time code to the embedded player application. The player application then retrieves the video, seeks to the requested time code (in-time), and begins playback. The user can interrupt the playback using standard VCR type controls on the player.*
>
> *The HTML code for an exemplary screen display is provided in the Appendix. Sheets A, B, C, D, E, G, H1, N and P are hereby incorporated by reference in their entirety and can be found in the USPTO file for application Ser. No. 09/134,499. Sheet A of the Appendix lists the directory names (clip and icons) and file names at a top level. Sheet B lists the files in the clip directory, while sheets C, D and E list the files in the icons directory. Sheet F lists the HTML code for the top level index.html file which provides the framework for the display shown in the browser window 916 (FIG. 17). Sheet G lists the contents of the topr.html file (as would be seen in the clip frame 912 (FIG. 17)). Sheet H lists the contents of the video_label.html file. Sheet I lists the contents of the video_mbase.html file. Sheet J lists the contents of the video_netshow.html file. Sheet K lists the contents of the video_noproxy.html file. Sheet L lists the contents of the video_ovs.html file. Sheet M lists the contents of the video_real.html file. Sheets J, K, L, and M may be used to provide the proxy video to allow different video formats to be displayed in the video frame 896 (FIG. 17). Sheet N lists the contents, including*

> *a set of keyframes and corresponding timecodes (as would be seen in the keyframe frame 904 (FIG. 17)), of the 000l.html file in the clips directory. Sheet P lists the contents, including a set of icons in a closed-caption text frame (as would be seen in the cc-text frame 908 (FIG. 17)), of the 000r.html file in the clips directory. The remaining sheets in the Appendix are alternate instances of the contents shown in exemplary sheets N and P. Of course, other programming languages besides HTML code could be used to implement hyperlinked output conversion.*

Here again, what is described is forming an HTML based version of the image shown in Fig. 2, and nothing is described that shows a *a picture content iconic region having icons representing pictures according to predefined content categories and picture metadata* as is suggested in the Office Action.

__3. a graphical browser region having and indicia of graphical browsers utilized by the GUI (Fig. 17) and further having a plurality of display levels linked to the main display level via one or more icons (Fig. 17, column 13 lines 1-33);__

Fig. 17 has been discussed in greater detail above. As noted above, as described at columns 12 lines 60-65 and columns 13 lines 1-32 Fig. 17 shows the following:

> *The Output Filter Manager 560 (FIG. 8) may utilize a HTML output filter 564 in one embodiment. Referring to FIG. 15, elements of FIGS. 1, 2 and 9 are shown together as utilized in generating HTML output. The user may invoke a GUI command such as the "Save-As" command on the "File" menu 553, which in turn provides a list of output filter choices (HTML, Real Networks SMIL, XML, custom, etc.). When the HTML filter 564 is invoked, it accesses the metadata in the Metadata Track Index Manager 530 and processes it into HTML form in a browser window 916 (FIG. 17), which also involves keyframe images in a keyframe frame 176 (FIG. 2) or 904 (FIG. 17), and the digital video 142 (FIG. 1) or as seen in a video frame 896 (FIG. 17). For instance, hyperlinks may be formed from displayed keyframes to video sequences. The digital video 142 may or may not be served by a content server 140. For instance, it could be a simple file on the file system of the client computer or, say, a networked mass storage device visible to the computer.*
>
> *Some key features of the Video Cataloger HTML output are:*
>
> *a. The HTML files used to generate the display in the browser window 916 (FIG. 17) are completely stand-alone, internally linked HTML, such that no Web server is required. Exemplary HTML files are provided in the Appendix and are described in conjunction with FIG. 17 below.*

> *b. It incorporates play-back of digital video 142 from a file or from a video server 140. That is, the digital video may be streamed directly to the browser, or it may simply be played from a local file on disk. The stand- alone aspect is strengthened when the digital video is a local file. This way, all of the content (HTML, keyframes, digital video) could be packaged up, compressed, and e-mailed to someone.*
>
> *c. All metadata is cross-referenced/cross-linked based on time-codes.*
>
> *d. Digital video is independent of the HTML representation—any digital video source can be linked into the playback frame.*

Thus, as is shown in these lines the window provided in Fig. 17 is a completely stand alone HTML browser window with video clips linked to the HTML browser window such that the video clips can be accessed without need to access a web server. Notably, as is described in column 14, lines 27 – 40, the video clips are presented in a video clip area 906. Thus, merely selecting a video clip from one of the displayed video clips simply causes the video clip to be presented in video clip area 904 of the same display level. Further, column 14, lines 47-51 explicitly states that it provides HTML code "for an exemplary screen display as is provided in the appendix." There is no apparent discussion in this section of forming more than one such screen display thus, Fig. 17 does not appear to stand for the proposition that Jain et al. describes a display *further having a plurality of display levels linked to the main display level filed one or more* as is suggested in the Office Action.

4. a picture grouping iconic region indicating files containing pictures in a database (Fig. 17) for picture grouping.

Jain et al. shows an area 904 of the HTML image of Fig. 17 wherein thumbnail images of so-called key frames extracted from video streams are positioned. These images can be clicked upon to cause a video clip associated therewith to be presented in area 896. What is displayed in this region are key frames extracted from the video. There is simply no discussion of key frame area 904 as containing icons, that such icons represente groups of pictures in a database, or that the icons are grouped in any particular manner such as being based upon content or metadata.

Thus, Jain et al. does not appear to support any of the above described points made in the Office Action of September 22, 2003. To reject claims in a patent application under *35 U.S.C.S. § 103*, an examiner must show an unrebutted prima facie case of obviousness. In the absence of a proper prima facie case of obviousness, an applicant who complies with the other statutory requirements is entitled to a patent. As the case made in the Office Action of September 22, 2003 relies upon Jain et al. to establish the points described above, and the points have now been rebutted, the applicants resectfully submit that claim 1, and all claims that depend upon claim 1 now stand ready for allowance.

Conclusion

The prima facie case of obviousness made in the Office Action of September 22, 2003 stands rebutted. Accordingly, all claims are in a condition for allowance, prompt notice of which is earnestly solicited.

Respectfully submitted,

Roland R. Schindler II
Attorney for Applicant(s)
Registration No. 40,802

RRS/rgr/tt
Rochester, NY 14650
Telephone: 585-588-2736
Facsimile: 585-477-1148